Europäisches Patentamt

Eur pean Patent Office

Office eur péen des brevets

EV

Bescheinigung

Certificate

Attestation

09/889512

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. **Patent application No.** **Demande de brevet n°**

99304784.4

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

Hatten

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 24/03/00
LA HAYE, LE

Europäisches Patentamt

Eur pean Patent Office

Office eur péen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.: Application no.: Demande n°:	99304784.4	Anmeldetag: Date of filing: Date de dépôt:	18/06/99

Anmelder:
Applicant(s):
Demandeur(s):

BRITISH TELECOMMUNICATIONS public limited company
London EC1A 7AJ
UNITED KINGDOM

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:

Translation

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

G06F17/28

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

TRANSLATION

This invention relates to automatic language translation.

Machine language translators accept input text in a first natural language (the source language) and generate corresponding output text in a second natural language (the target language). Such translators may be classified into two types; those which use a set of translation rules for each possible pair of source and target languages, and those (relatively rare) interlingual systems which translate from the source language into a language independent (interlingual) form, and then from this language independent form to the target language.

In the system described in our earlier application number PCT/GB98/02389, rules specifying the complements which each verb of all source and target languages could take were present, and were stored with pointers from the corresponding verb entries in a lexical database. These rules also specified the mapping between the complements and the roles (e.g. agent or patient) corresponding to them.

The roles were assigned in a relatively simple way, with the subject of the verb always the active role (agent) and the object the passive role (patient). Abstraction rules then dealt with the necessary changes to the role in unusual cases. Complements attached by prepositional phrases would not have roles; these were assigned by abstraction rules.

The rules needed to be hand written, and since this was required on the order of one per verb per language, the effort was considerable and the results were not consistent.

In the present invention, by way of contrast, all possible roles which a given verb (or other word capable of taking complements) can have are represented, at a semantic and interlingual level, in role set records, and the mapping between these roles and the complements which can be taken by each verb in each language are captured by lists of alternations (i.e. allowable word orders of complements of the verb) stored in alternation class records, which may each be used by multiple verbs.

Since the presence of prepositions and other verb form irregularities is captured in a relatively small number of alternation records, a relatively small number of parsing and abstraction rules to replace such prepositions and other features on detection of their occurrence can be employed.

Other aspects and preferred embodiments are as described in the following description and claims.

Embodiments of the invention will now be illustrated, by way of example only, with reference to the accompanying drawings, in which:

Figure 1 is a block diagram of the language translation apparatus according to a first embodiment;

Figure 2 is a block diagram showing in greater detail the processes present in a client terminal forming part of the embodiment of Figure 1;

Figure 3 is a block diagram showing in greater detail the processes present in a server forming part of the embodiment of Figure 1;

Figure 4 is a block diagram showing in greater detail the subprocesses present within a translation process forming part of the embodiment of Figure 3;

Figure 5 is an illustrative diagram showing the formats through which text passes during the translation process of the embodiment of Figure 1;

Figure 6 is a block diagram showing the databases maintained within the server of Figure 1;

Figure 7 is a schematic diagram illustrating the word structure produced after text pre-processing in the embodiment of Figure 1;

Figure 8 is a diagram illustrating the entity/relationship semantic structure produced after parsing in the embodiment of Figure 1;

Figure 9 is a flow diagram showing schematically the operation of the server of the embodiment of Figure 1;

Figure 10 is a diagram illustrating the data stores present in the embodiment;

Figure 11 is a diagram illustrating the relationship between records in the embodiment;

Figure 12 is a flow diagram illustrating a compilation phase of writing records to each word store of Figure 10;

Figure 13 is a diagram illustrating the relationship between records during the process of Figure 12;

Figure 14a illustrates a pre-abstracted. language-specific structure, and

Figure 14b illustrates the structure after application of an abstraction rule,

Figure 15 is a flow diagram illustrating the abstraction rule operation;

Figure 16 (comprising Figures 16a and 16b) is a flow diagram illustrating the process of generating the data used in the present embodiment; and

Figure 17 illustrates a role set record of the embodiment.

Background to Embodiment

For ease of reading, features of PCT/GB98/02389 are reiterated here; the whole of the description is incorporated herein by reference.

Referring to Figure 1, the present invention may be employed by a client terminal 100a connected via a telecommunications network 300 such as the Public Switched Telephone Network (PSTN) to a server computer 200. The terms "client" and "server" in this embodiment are illustrative but not limiting to any particular architecture or functionality.

The client terminal comprises a keyboard 102, a VDU 104, a modem 106, and a computer 108 comprising a processor, mass storage such as a hard disk drive, and working storage, such as RAM. For example, a SUN™ work station or a Pentium™ personal computer may be employed as the client terminal 100a.

Stored within the client terminal (e.g. on the hard disk drive thereof) is an operating control program 110 comprising an operating system 112 (such as Windows™), a browser 114 (such as Windows Explorer™ Version 3) and an application designed to operate with the browser 114, termed an applet, 116. The function of the operating system is conventional and will not be described further. The function of the browser 114 is to interact, in known fashion, with hypertext information received from the server 200 via the PSTN 300 and modem 106. The browser 114 thereby downloads the applet 116 at the beginning of the communications session, as part of a hypertext document from the server 200. The function of the applet 116 is to control the display of received information, and to allow the input of information for uploading to the server 200 by the user, through the browser 114.

Referring to Figure 3, the server 200 comprises an operating program 210 comprising an operating system 212 such as Unix™, a server program 214 and a translator program 216. The operating system is conventional and will not be described further. The function of the server program 214 is to receive requests for hypertext documents from the client terminal 100a and to supply hypertext documents in reply.

Specifically, the server program 214 initially downloads a document containing the applet 116 for the client terminal 100a. The server program 214 is also arranged to supply data to and receive data from the translator program 216, via, for example, a cgi.bin mechanism.

The function of the translator program 216 is to receive text from the client terminal 100a via the telecommunications network 300 and server program 214; to interact with the user as necessary in order to clarify the text; and to produce a translation of the text for supply back to the user (in this embodiment).

Figure 4 shows the component programs of the translator 216. It comprises a number of sections; one for each language, of which only a first section 220, relating to a first language (LANG1) and a second section 230 relating to a second language (LANG2), are shown for clarity. Each language section comprises the following subprograms or modules:

1) A text pre-processor (221, 231)

2) A source language parser (222, 232)

3) A source language abstractor (223, 233)

4) A target language de-abstractor (224, 234)

5) A target language generator (225, 235)

6) A target language text post-processor (226, 236)

The functions of each of these components will be discussed in greater detail below.

Figure 5 illustrates the stages of translation according to this embodiment.

A source language text document (stage A) is received by the translator from the client terminal 100a.

After operation of the text pre-processor stage (221), the result is an expanded source language text document (stage B). The operation of the pre-processor is to replace contracted forms of words (such as "he's" in English, or "j'ai" in French) with their non-contracted forms.

After operation of the source language parser 222, stage C of Figure 5 is a language-specific semantic structure which represents the input text as an encoded entity-relationship graph, where the entities are semantic categories corresponding to the words (in other words, identifying the nouns, verbs and so on), and the relationships are data relating the entities together (e.g. to indicate those which are the subjects or objects of others).

After operation of the source language abstractor 223, the result at stage D is a further semantic structure D, similar to the language specific semantic structure produced at stage C but indicating additional relationships and data which substitute the language-specific meanings of some of the structures represented within the semantic structure C with abstracted structures.

For example, a phrase such as "My name is David" input as source language text could be represented within a parsed semantic structure by data indicating ownership of the name by the individual first person, and an attribute of the name being that it is "David". This is a grammatically correct expression, from which French or German text could be generated by a suitable generator such as 235.

However, whilst grammatical French or German would be produced, the meaning would be unclear, since in French the equivalent phrase is "I call myself" ("je m'appelle") and in German the equivalent phrase is "Ich heiβe" (which is equivalent to "I am called" in English, but for which English lacks a corresponding verb). Accordingly, the source language abstractor 223 recognises, within the parsed semantic structure, the occurrence of structures which are not directly translatable, such as structures involving personal names in this example, and replaces those structures with additional data representing them.

~~Accordingly, the abstracted semantic structure produced at stage D of Figure 5~~ corresponds to a representation of the input text but with the replacement of specific constructs which are known not to meaningfully translate into one or more other languages (whether or not those languages are represented by sections within the translator 216).

The abstracted semantic structure produced at stage D is an interlingual form which is unambiguous in relation to each of the target languages which the system is capable of translating into. That is to say that the interlingual form corresponds uniquely with a language-specific semantic structure in each of the target languages.

The abstracted semantic structure, or one of the abstracted semantic structures, produced by the abstractor in stage D is then passed to the de-abstractor 234 of the target language, which comprises a series of rules which test for the presence of the additional structures inserted by the language abstractor 223, and translate them into the form used in the target language. For instance, in the example given above, the abstracted naming operation would be converted, in French, into "je me appelle" (I call myself). The result is then, at stage E, a semantic structure equivalent to the language-specific semantic structure at stage C but in which the semantic substructures corresponding to phrases or expressions in the input text which would give rise to translation difficulties have been replaced by appropriate substructures in the target language. This structure forms the input to the target language generator 235, which generates a corresponding target language output text (stage F), and therefore applies the reverse process to the parsers 222, 232.

Finally, the generated output text at stage F is contracted by the text post-processor 236 which takes the generated text and contracts relevant parts of it. In the above example, "je me appelle David" would be contracted to "je m'appelle David". Other minor text processing operations, such as adding capital letters at appropriate places (for example at the beginning of each sentence), and providing the correct spacing between words, are also carried out.

Referring to Figure 6, the server 200 stores data for use by the parser and abstractor in each language. This data comprises, for each language, a grammar rules database (227, 237) and an abstraction rules database (228, 238). Also present is a multilingual lexical database 240. The lexical database 240 stores an entry for each concept represented by a word in any language represented within the translator program, the entry pointing to corresponding work entries in each of the languages within which equivalents to that word exists, which give for each of the text in the language concerned; the type of lexical element represented by the word (e.g. whether it is a noun, a verb, a pronoun, an adjective and so on); data on the manner in which the word is inflected, if at all, in each language, and various other data.

The grammar rules stored within each grammar rules database (227, 237) represent, for the corresponding language, the ways in which words of that language may combined.

The operation of this embodiment will now be disclosed in greater detail with reference to Figures 7-11.

Referring to Figure 9, in a step 402, text is received from the client terminal 100a. In a step 404, the input text is expanded. As a first step, the start and end of each possible word in the text is located by detecting spaces and punctuation, so as to result in a stream of possible words. As a second step, any contracted words (such as "j'ai" in French") are expanded to replace them with full words (in that example, "je ai"). At the same time, the text pre-processor locates and flags special text items such as proper names, dates, times, sums of money and so on.

At this stage, there may be several possible expanded strings of words that could match each contracted string of word. All such possibilities are retained as alternatives.

Next, each word is looked up in the lexical database 240. At this stage, words which are not recognised but are closely matched to others in the source language (that is, the language of the input text) are replaced by all those for which they are a close match, as in the manner of a conventional spell checker.

If, after spell checking, any words have not been recognised (step 405) then a query is transmitted back to the user, comprising a text message saying, for example, "The word (unrecognised word) has not been recognised. Please check the spelling, and resubmit this word or a synonym". This query is then transmitted to the client terminal 100a in step 406.

The result of this pre-processing is therefore that the expanded text (stage B of Figure 5) is no longer necessarily a linear sequence of words but may, as shown in Figure 7, comprise a network or lattice of words.

Figure 7 indicates such a network in which the second word, originally B, has been replaced by two possible alternatives (either alternative spellings or alternative expansions) B1 and B2, and the third word C has been replaced by three possible alternatives C1, C2 and C3. There are thus now six possible routes through the network of words.

The text of each word in the network is now replaced by a reference to the corresponding entry in the lexical database 240. If a single word (such as "bank" in

English) has two different entries in the lexical database 240 corresponding to different meanings (which would be translated into different words in a target language), the word is replaced by each possible entry in the lexical database 240. For convenience, rather than using references to the entries in the lexical database, the syntactic category information for each word (i.e. whether it is a noun, verb etc.) may be retained within the network, and a table relating each network position to the corresponding entry in the lexical database 240 may be separately stored for later use.

On each occasion where a single word in the source language is given as the translation of several different lexical entities in the database 240 (corresponding to several different words in one or more of the target languages), a reference to each of these is included within the processed text lattice of Figure 7.

Further details of parsing are given below.

Next, the network of nodes (each corresponding, as noted above, to one of the entries in the lexical database 240 and being represented by the syntactic category of that entry) is processed by the source language parser program, which, for each word, applies the rules within the grammar rules database 227 which are applicable to words of that type.

Thus, for example, referring to Figure 8, suppose that the English text contained the phrase "the dog saw the cat". The word "the" is the definite article, and a rule within the grammar rules database 227 indicates that it can be followed by the noun to which it refers. Thus, the circle D1 indicating the first occurrence of determiner "the" is linked by this rule to the next circle N1, representing the following noun "dog", and the circle D2, representing the second occurrence of determiner "the" is linked by this rule to the circle N2 for the following word, which is the noun "cat".

The rule for the active form of the verb "to see" indicates that the verb may be preceded by the seeing "agent" entity (in this case "the dog") and followed by the patient entity (in this case "the cat").

Thus, after parsing, the parsed semantic structure (stage C of Figure 5) is represented, for each sentence of the input text, by one or more structures comprising references to entries in the lexical database 240 (the circles in Figure 8) and pointers linking them together (the lines in Figure 8). In the PROLOG computer language, the topological structure of Figure 8 may be represented as

```
[
        A^det(def,s,_,third),A^e(dog,[]),P^det(def,s,_,third),P^e(cat,[]),
        E^event(see,fin,past,[]),E^A^r(agent,[]),E^P^r(patient,[])
]
```

In the foregoing, it will be noted that the unifying variables A and P are the links which unify the first occurrence of "the" with "dog" and the second occurrence of "the" with "cat". The verb "see" is linked by an agent relationship and a patient relationship with the terms linked by the relationship A (i.e. "the dog") and the terms linked by the relationship P (i.e. "the cat").

The verb is recorded as an event ("event"), and is linked to the lexical entry in the lexical database 240 for the word "see" and is indicated to be the finite form ("fin") in the past tense ("past").

The word "the" is recorded as a determiner, being the definite article ("def"), single rather than plural form ("s"), having neutral gender ("_") and referring to the third person ("third"). The terms for "dog" and "cat" are indicated to be entities ("e"), and have a reference to the corresponding word entry in the lexical database 240.

Thus far, other than the target-language dependency, the parser is not dissimilar to known, technically and commercially available products. Further information on suitable chart-parsing techniques which may be used will be found in James Allen, "Natural Language Understanding", 2nd Edition, Benjamin Cummings Publications Inc., 1995.

In other respects, the parser may be as described in PCT/GB98/02389.

Having thus parsed the text (step 410), the abstractor 223 then accesses the abstracting rules database 228 to locate those source language phrases which may give rise to translation difficulties. The abstraction process is recursive, insofar as once one abstraction rule has been applied to the parsed text, the entire set of abstraction rules is referred to again when processing the partially abstracted text to identify another abstraction rule to be applied, repetitively until none of the abstraction rules in the set can be applied.

Thus, in step 412, the abstractor 223 tests each structure generated by the parser, and where one or more of the abstraction rules is applicable, converts the detected structure to the alternative form recorded within the rule. As explained, this test is recursive such that the same rule may be applied at different stages of an abstraction process in which a structure generated by the parser is converted to the interlingual structure.

After operation of the abstractor 223, the ideal result should be a single, complete interlingual structure. If the structure is incomplete (that is to say, it was not possible to relate together all the words using the grammar and the abstraction rules) then successful translation will not be possible. If more than one possible structure is produced, then the input text is considered ambiguous since it could result in more than one possible translation in at least one of the target languages. If either of these conditions is met (step 414), a query is transmitted to the user (step 406).

In greater detail, the problematic points within the semantic structure, corresponding to incomplete or ambiguous meanings, are located, and the portions of the input text relating to these are formulated into a message and transmitted back to the user for display and response by the applet 116, with a query text which may for example say "the following text has not been understood/is ambiguous."

In a preferred version of the present embodiment, the de-abstractor and generator 224, 225 corresponding to the input (source) language are employed (as described in greater detail below) to generate a source language text for each possible semantic structure where two or more such structures exist, and the query also includes these texts, prefixed with a statement "one of the following meanings may be intended, please indicate which is applicable:"

In this case, the message transmitted to the user in step 406 comprises a form, with control areas which may be selected by the user at the client terminal 100a to indicate an intended meaning for the ambiguous words or phrases detected within the input text.

If no such ambiguities are detected, or after all such ambiguities are resolved (step 414), the single, unified, interlingual semantic structure produced by the abstractor 223 is then passed to the target language de-abstractor 234 for the or each target language into which the text is to be translated. The de-abstractor 234 accesses the abstracting rules database 238 and, on detection of any of the substituted forms (for example "I sit") substitutes the normal form for the target language (in this case, "I sit myself" in French or "I am sitting" in English). The de-abstracted structure is then more idiomatically correct in the target language than was the semantic structure produced by the parser.

Next, in step 418, the target language generator program 235 accesses the target language grammar rule database 237 and the lexical database 240 and operates upon the de-abstracted semantic structure to generate output target language text.

The operation of the generator is essentially the reverse of that of the parser; briefly stated, it operates a chart-parsing algorithm (of a type known of itself) to take the components of the target language semantic structure generated by the de-abstractor, look up the applicable rules in the target language rules database 237, and assemble the corresponding words located from the lexical database 240 into a string of text ordered in accordance with the grammar rules, until a single stream of text which utilises all components of the semantic structure and obeys the grammatical rules is located.

After generating the output text stream, the text is post processed (step 420) to add a space before each word; capitalise the first letter in a sentence; add a full stop after the last word; contract any phrases (such as "je ai") which are capable of contraction; and reproduce any special forms of text (such as dates, amounts of money, and personal names), as appropriate for the target language.

The resulting formatted text is then formulated into an HTML (or text, or other suitable format) page, which is transmitted back to the user at the client terminal 100a in step 422.

On receipt of the translation result at the client terminal 100a, the page is displayed via the browser 114 and may be converted and stored for subsequent word-processing by the user.

First Embodiment

. In English, and in many other languages, a phrase involving a verb may have several different possible word orders, each of which is referred to here as an "alternation". This embodiment provides improved rules for dealing with the alternations which are associated with words (particularly verbs) which can take complements in several different orders (alternations). A description of alternations in English is to be found in "English verb classes and alternations", Beth Levin, Chicago Press 1993, ISBN 0226475336.

In English, many verbs have a prepositional phrase as a complement; for example; the verb "give" may have a noun phrase and a prepositional phrase as compliments, as in the example "I give [the book] [to the girl]". The preposition may not be required in the equivalent phrase in other languages. For example, in the English phrase "they look for the ball", the preposition "for" is not represented in the French equivalent "ils cherchent la balle".

Others have a preposition-like participle attached - for example "bring in", in which the word "in" has no meaning except to modify the meaning of "bring".

Many verbs of attitude (i.e. verbs expressing states of mind) may have a reversible form. For example, the statement in English "I like the book" is equivalent in meaning to "the book pleases me", although there will in some cases be a subtle shift of emphasis. Both would be translated in Spanish, for example, as "el libro me gusta".

In each case it will be seen that the rules governing the use of the verb and the surrounding word order are quite language-specific, and therefore will need to have associated abstraction rules and de-abstraction rules. Unfortunately, to write separate abstraction rules for each verb is an enormous task for each language separately, and leaves open the risk that rules may be missed. It also leads to ad hoc and unsystematic development of the abstraction rules.

This embodiment therefore provides a new method of creating rules for handling alternations (particularly for verbs), and new methods of use thereof.

Data Structures

Referring to Figure 10, the data structures employed in the present embodiment will now be described.

Referring to Figure 10, the lexical database 240 comprises, for each language, a list 1241, 1242 of word entries or records. Each word record in each language list points to a concept or meaning record entry in a language independent meaning store 1240. Each record in the store 1240 contains meaning (i.e. semantic information) relating to a concept, described by the word entries which point to that record.

One suitable structure for such a meaning store (lexicon) is given in the WordNet (TM) lexical database, available from Princeton University, Princeton, New Jersey, USA or MIT Press Five Cambridge Center, Cambridge, MA USA, details of which are at http://www.cogsci.princetown.edu/~wn/.

Thus, word records in the word list stores 1241, 1242 of different languages are indirectly linked, in that they point to a common entry in the semantic lexicon 240, which is related to the meaning expressed by the words.

Referring to Figures 10 and 11, a plurality of word entries 240, 241, 242, 243, 244, in each word store 1241 in a given language of the lexical database 240, each contain a pointer to an alternation class record (702, 704, 706) in a corresponding one of a plurality of language-specific alternation class stores 1238, 1228 provided within respective grammar rules stores 228, 238 corresponding to each of the languages to be used.

The relationship is a many-to-one mapping. That is to say, many verbs (of the order of several thousand in English) map onto a relatively small number of alternation classes (of the order of 200 in this embodiment in English). Each word entry in the word store 1241, 1242 maps onto to only one alternation class record per language. Several lexical entries will share the same alternation class record. The significance of the alternation class records will be explained below.

In each of the alternation record stores 1228, 1238 (each said store being associated with a respective language), each alternation class record 702-706 is linked to one or more alternation records 708-722 by pointers.

Compiling the Word Entries

In this embodiment, the contents of the word stores 1241, 1242 are not fully populated until the apparatus is to be used. This results in a saving of memory space, since those word stores for languages which are not to be used in translation require less memory.

Accordingly, referring to Figures 12 and 13, a first set of base word entries (WORD 1 of Figure 13 for example) remain resident in the store 1241 at all times. Where the word is one which can take multiple alternations, the word entry is linked by a pointer to an alternation class record (ALT CLASS of Figure 13) in the alternation store 1238 for the language concerned. This record is linked to the alternation records (ALT 1, ALT 2 of Figure 13) of the alternations which that word (and others sharing its alternation class) can take.

A program, operated prior to translation, performs the process of Figure 12. In a step 1402, the class record for a word is read, and a first alternation record is selected in a step 1404. In a step 1406, the program creates a new lexical entry (WORD 2) for the same word as that for WORD 1, which incorporates a pointer to the alternation class record, and stores the alternation as a list of possible complements in an order.

If there are more alternation records (step 1408), the process of step 1404 on is repeated for the next alternation (step 1410). If not, the next pre-existing word record in the store 1241 is selected (step 1414) and the process of step 1402 is repeated until all word entries have thus been expanded to an entry for each alternation (step 1412).

Referring to Figures 10 and 17, also provided in this embodiment is a role set store 740 storing a plurality of role set records, one of which is indicated (as 730) in Figure 17. Each role set record stores a plurality of role data, shown as R1, R2 and R3 in Figure 17. In the present embodiment, there are on the order of 15-20 role set records in total in the role set record store 740.

The entry for each event concept in the meaning store 1243 (e.g. each entity corresponding to an event or verb in a word store 1241, 1242) or other concept which can take an alternation in some language, is linked by a pointer to one of the role set records. It will therefore be clear that, since there are many thousands of such verb or event records in each word store of the lexical database 240, there is many-to-one mapping of lexical database entries to role set records.

Each of the alternation class records in the alternation class store 1241, 1242 for a language is linked by a pointer to one of the role set records 740. Since the number of role set records is substantially smaller than that of alternation class records in each language (e.g. by an order of magnitude), this too is a many-to-one mapping.

The significance of the data stored in each record will now be explained.

For each word representing an event in a language (e.g. a verb in English) the lexical database 240 stores a record for one or more semantic concepts which correspond to that word. For example, the word "give" in English may b associated with several different concepts, including "give to" and "give up". Each of these provides a different meaning for the word.

Within the word list for each language, separate entries are provided for each verb meaning, and in particular for each meaningful verb/preposition pairing. Thus, the verb "look" in English has one entry, but then "look for", "look at" and so on have separate entries, pointing to different concept records in the lexical database 240.

Associated with each such event concept, then, are one or more roles; that is to say, people or objects taking part in the event. For example, corresponding to the phrase "they looked for the ball", are an active role or agent ("they" - the people who looked) and a passive role or patient ("the ball" the thing that was looked at). Other events may involve more parties, for example, "[she] gave [it] [to him]" involves a donor, a recipient and an object. These roles are the same regardless of the word order employed (e.g. "he was given it by her") or the verb form used (e.g. "the book pleases me" and "I like the book"). They may vary slightly between languages, since in some languages certain roles may be inferred, but will be similar across all languages. Thus, the role set record stores for a concept all roles which might be used in any language for that concept. They do not necessarily correspond to the subject and object of a verb, since these vary between active and passive forms of a verb.

Many different events (i.e. verbs) can be described using the same roles. Thus, according to the present embodiment it has been determined that a relatively small number of role sets (each represented by a role set record 730) can be provided, one or other of which will provide the necessary set of roles for all events in any language (see e.g. M A K Halliday "An introduction to Functional Grammar" (2nd Ed, 1994, Edward Arnold), ISBN 0340574917). The number of roles in each role set record, and their identities, differ from record to record. The role set records therefore represent a language-independent data structure linked to those concept records of the lexical database records which are for event concepts. Examples of (almost all the commonest) roles are

- For material "action" - agent and patient. This is the largest category with verbs of most actions.
- For "behavioral" events - a behaver. These are generally physiological events such as yawning.
- For "perception" and thinking events - senser and phenomenon. Examples are Seeing, hearing, feeling.
- For "verbal" events - sender, message, and recipient. Examples are Saying, writing etc.
- For "relational" (stative) verbs - carrier, attribute or token, and value, depending on the kind of verb. For example, "the house is big", "John is the leader"
- For "existential" (there is/there are) - existent (i.e. that which exists).

Associated with each role field in this embodiment is a restriction field, the use of which will be discussed further below.

Each alternation class record 702-706 contains a record of the language it is relevant to, together with a name field naming the class record (for example, "reflexive verb"), and is linked by pointers to its role set record and alternation records.

Each alternation record 708-722 comprises a name field; a pointer to the alternation class record to which it is linked; the syntactic category (e.g. verb) of the alternation; and a list of terms each of which maps a role (which is one of those listed in the role set record of the alternation class to which the alternation record belongs) to a syntactic category (e.g. noun, preposition, and so on). Conveniently, the order of the mapping fields is significant. For a verb, the first mapping field is taken to indicate the subject of the verb (when it is present in the active form) and the remaining mapping fields indicate the complements of the verb in order.

Of the names for each alternation, one alternation of each alternation class is always named "normal"; this alternation will specify the word order most commonly used in the language concerned. The names of the other alternations in each alternation class will specify the conditions under which that alternation is used; for example, "polite", "formal", "stressed".

The declaration of an alternation (in PROLOG) may for example be as follows:

```
alternation(hold_onto_relate, relate, normal, eng,
        v:[],
        [
                agent->np:[case=nom],
                        patient->pp:[pform=onto]
        ]
).
```

Here 'hold_onto' is the name of the alternation class, 'relate' is the name of the role set record, 'normal' is the name of this alternation within the alternation class, 'eng' is the language code 'v:[]' is the syntactic category (verb).

Where an alternation specifies a prepositional phrase, the identity of the preposition is stored in the alternation (above, for example, "onto").

In some cases, where a particle such as the preposition-like particle "in" in the phrase "bring it in" can accompany the word, it is given a special "null" role, indicating it has no separate semantic significance. On compilation, this role is listed in the word record created from the alternation, and is hence required to be present if that alternation is to be detected during parsing, but no semantic terms are thereby introduced during translation.

Thus, the data stored in a set of alternations (for example, the set of different alternations in which the verb "hold" can be used with a preposition "onto" to express the concept "hold onto" in English) can be used to locate, within a given phrase involving that verb, which of the surrounding words or phrases occupies which role in relation to the verb.

This information can then be used to re-generate a corresponding phrase for the same concept expressed in the target language, since the set of roles is defined (in a language independent fashion) by the frame to which both the source and target language alternation classes point, and from which both were derived.

In use, the present embodiment operates as follows.

Parsing

In this embodiment, parsing is performed as described above, to generate the language-specific parsed semantic structure.

During parsing, as mentioned above, each expanded word in the document is looked up in the word store 1241 of the source language. The word is then replaced by a reference to the word entry or entries corresponding to it, for subsequent use.

Where the word is a verb (for example, "give") which can be used in several different senses, several different entries will be found in the word store, (e.g. corresponding to "give to", "give up" and so on). Each expresses a different concept, and therefore points to a different concept entry in the lexical database 240.

Each also points to an alternation class record; the two entries may point to different alternation class records.

Further, where the word can take several alternations, a separate word entry for each alternation will have been created, as described above.

Thus, after all words have been looked up in the source language word store 1241, the parser uses the orders defined in the word records, together with the rules stored in the rule store, to attempt to create paths through the word lattice of Figure 7.

Since only one of the alternations will actually be present, those word entries corresponding to alternations which have a complement order other than that detected to be present will be rejected during parsing.

Thus, after operation of the parser, the parsed semantic structure will include one or more identified alternations for each event term located in the input document, the alternations being identified where the syntactic categories surrounding the identified event in the source document match those in the order specified in the alternation.

Since the alternation record maps the syntactic categories surrounding the event to their roles by their order of occurrence, then as shown in Figure 8, the parsed semantic structure identifies the role performed by each syntactic category around the event.

Abstraction

At this stage, referring to Figure 14a, the semantic structure will still include any prepositions originally present; for example, the phrase "to the girl" will be identified as the patient entity in the phrase "he gave the ball to the girl", with "to" identified as a preposition. Also, in the case of verbs with prepositions and some other types of verb (for example, verbs in the passive form) the roles identified during parsing will not be language independent.

Accordingly, an abstracting rule is provided which, in the abstracting phrase, identifies each word term in the parsed semantic structure (step 1002), looks up the corresponding word record, and from that, accesses the corresponding alternation class record (step 1004), and thence the alternation record (step 1006) corresponding to the alternation used to generate that word record.

If the alternation indicates that a preposition phrase is present in the source language (step 1008), then the abstraction rule deletes the entry for the preposition from the parsed semantic structure (step 1010), so that instead of pointing to the prepositional phrase "to the girl" as the object (or some other language dependent role), the event term points to the phrase "the girl" which followed the preposition.

Finally, in step 1012, data recording the language-independent role assigned to that prepositional phrase in the alternation record (here, "patient", as shown in Figure 14b) is assigned to the phrase which followed the preposition. The abstraction rule then proceeds in similar fashion until all terms in the parsed semantic structure are processed.

Other word forms where the assignment of complements to roles is language dependent can likewise be detected and amended.

Since the abstraction rules can access the alternation records, and since the alternation records are derived from a language-independent set of roles shared by all translations of the verb being abstracted, the abstraction rules can identify the complements corresponding to each language-independent role and label them correctly where the original role assigned depended upon the source language.

Conveniently, during abstracting, the references from each term in the parsed structure to its source language word entry are replaced by references to the corresponding meaning term in the language-independent meaning store 1243. At the same time, "register" or "tone" data indicating the tone (for example, "normal", "formal" or "informal") of each word entry where several correspond to the same meaning entry with different tones) is stored with the reference to the meaning entry.

De-abstracting

In de-abstraction, the meaning entry references of the terms of the interlingual structure are each looked up in the meaning store 1243, and a word entry (in the target language word store 1242) with corresponding tone data to that stored for each term is selected. The roles present for each event term are then compared with those for each alternation recora of the alternation class pointed to by the selected word entry, and the best- matching alternation record is selected.

The language-independent roles of the interlingual structure are then replaced, where necessary, as specified by that alternation (for example, where the verb in the target language has its roles reversed relative to the source language).

References to the selected target language word records are then substituted for the references to meaning records in the interlingual representation of the document.

Generation

Generation of the target language text then proceeds using the selected word records.

Generation of alternation data

The process of creating the data records used in the present embodiment will now be described.

Conveniently, the input and editing processes may be performed using the terminal 100 to access the server 200, from which the lexical database 240 and other records are read and to which they are written, via a browser program providing a graphical user interface into which data may be input and edited.

In a step 2002, the role set records (or, at any rate, most of them) are created by the user, and each meaning entry in the lexical database 240 which can have multiple alternations is assigned to one of the role sets (as mentioned above, there are typically 15-20 such role sets).

In a step 2004, a first language word store 1241 employed in the translation system (either as a source or a target language or both) is selected. For each language word store, the word entries will already have been assigned pointers to corresponding meaning entries in the meaning store 1243.

In a step 2006, a first event entry in the word store is selected.

Next, in a step 2008, the alternation classes associated with that role set, in the language concerned, are displayed. If the event is the first event in that role set to be considered, there will be no alternation class displayed.

The data displayed (step 2106) for the alternation class is the list of alternations of the alternation class, displaying for each the role-complement mappings present in that alternation.

If no suitable class exists yet (step 2010), a new class is created (step 2012). Usually, a suitable class will exist already. In either case, in step 2014, the event is allocated to the alternation class it matches or the class which has newly been created.

If no alternation have yet been defined for the class, a template alternation listing the roles present in the class, in some order, is displayed and the user edits the display to re-order the roles into the desired order, add prepositions as desired, and so on.

If (step 2018) the list of alternations does not match those known by the inputter to exist for the word in the language concerned, then new alternations are created (step 2020) in the same way and added to the alternation class (step 2022).

If the last event in the language has not been reached (step 2024) the next event is selected (step 2026) and steps 2006 onwards are repeated.

If there are more languages to process (step 2028) the next is selected (step 2030) and the process returns to step 2004.

When all languages have been processed (or at any other desired end point) the data input is stored (step 2032) as alternation class records (702-706), alternation records 702-722, and role set records 730.

As mentioned above, it is typically found that relatively small number of role set records and a larger, but still small, number of alternation classes (of the order of a few hundred) per language required. The small number of role set records results from the relatively small number of different roles which can be played in events, and the relatively small number of alternation classes results from the same fact, and also from the tendency of many verbs to behave similarly.

The number of alternation records will vary from class to class and from language to language. The number of records will increase with the mutability of the word order in each language and with the irregularity of word orders between different verbs.

Role preference data

As stated above, associated with each of the role fields in the role set records 730 may be a role preference field.

For example, the lexical database 2040 may be hierarchically arranged, as described in PCT/GB98/03774 filed 16/12/98 priority 17/12/97, so that, for example an entry for "computer" points to a hierarchically higher entry for "electrical equipment" which in turn points to a hierarchically higher entry for "man made artefact" which in turn points to a hierarchically higher entry for "artefact" and thence to an entry for "entity".

Where the lexical database is hierarchically ordered in this manner (or, with greater difficulty, even where it is not organised in this manner) the preference field associated with each role may be set to point to a corresponding entry in the lexical database.

Thus, for example, certain types of activity are performed only by living creatures, and some only by people, so that the preference data for the "agent" role in these cases will be set respectively to point to the entries in the lexical database for "living creature" and "person".

Indirectly, through the hierarchical arrangement of the lexical database 240, the preference data therefore also points to all the hierarchically lower instances of those general classes which are stored in the lexical database.

~~The usefulness of such preference data is seen where the output produced by~~ the parser is either incomplete or ambiguous. For example, if a given part of a document can be parsed to give two meanings, allocating different words or phrases to different roles, the ambiguity may be resolved during abstracting.

Each possible such parsed structure is matched to locate the corresponding alternation record, from which the presumed roles of each part of the parsed structure are determined. The role set record for the alternations is then examined, and it is determined whether the entities allocated to each role correspond to those specified in the preferences. The meaning in which the entities correspond more closely the specified preferences is then selected as likelier to be correct.

Similarly, where incomplete input text is located by the parser so that complete parse cannot be performed, but nonetheless it is possible to locate for example a verb and preposition so that the correct role set record can be located, a comparison of the preference data stored for the roles in the role set record with the entries in the lexical database for the text surrounding the verb may suffice to complete the parse by allocating roles to the text present.

Other Embodiments and Variants

Other words than verbs can benefit from the invention; it may, for example, be used to compile multiple word entries for words which can change their form - e.g. adjectives which have an adverbial form. Each alternation record within a class can have a different syntactic category (e.g. adverb and adjective) and the record can thus be used to specify whether the derivation of a different word form can take place, and what the feature changes should be.

Although it is preferred to retain the role set records where they store role restriction data, if such data is not used in translation then the role set records need not be present in the translator, being only used to derive the alternations consistently between different languages as described above.

Although the above embodiments accept a text document, a speech recognition front-end is also possible, or an image scanner with optical character recognition could be employed.

Although the above described embodiments describe a translation system, in which the target language text is generated, it will be understood that it would be possible with advantage to utilise the interlingual language structure generated for other purposes; for example, to provide a natural language front end or input routine for control of a computer or other equipment. Accordingly, such other uses of some aspects of the invention are not excluded.

Although adaptation to the intended target languages by limiting the search within the lexical database 240 to those words occurring in the source and those target languages has been described, it will be realised that it would also be possible to limit the operation of the abstractor, and merely to utilise those abstraction rules which remove language dependency in the source language which is not also present in the intended target languages.

In this case, each abstraction rule would similarly include a reference to those languages for which it was necessary, and only the necessary rules for the intended target language(s) would be used. Such an embodiment may prove useful as the number of target languages increases.

The foregoing embodiments are merely examples of the invention and are not intended to be limiting, it being understood that many other alternatives and variants are possible within the scope of the invention. Protection is sought for any and all novel subject matter disclosed herein and combinations of such subject matter.

CLAIMS

1. Translation apparatus for translating a document from a source language to an interlingual representation in which it can be transformed into one or more of a plurality of target languages, each of said languages including words corresponding to events, in which entities play predetermined roles, comprising

means for storing, for each word representing a said event, in each said language, an alternation set of role orders each listing the correspondence between said roles for that event and the complements by which it can validly be represented in that language;

means for locating, in said document, one or more words representing an event, and the complements representing roles associated with it, and representing said phrase in a language-dependent semantic structure; and

means for replacing said language-dependent semantic structure with an indication of the language-independent roles represented by said complements using said role orders

2. Apparatus according to claim 1, in which said events are defined by a verb in said source language, and said means for replacing are arranged to locate a preposition occurring with said verb, and to replace said preposition with an appropriate role.

3. Apparatus according to any preceding claim, which is arranged to generate a document in said target language in accordance with a said alternation set in the target language, by selecting one of the role orders thereof.

4. Apparatus according to any preceding claim, wherein a single said alternation set is stored in respect of multiple said events.

5. Apparatus according to any preceding claim, further comprising means for storing, for each event, a language-independent indication of the roles which may be associated with that event in any of the said languages.

6. Apparatus according to any preceding claim, wherein a single said indication is stored in respect of multiple said events.

7. Apparatus according to claim 8, in which multiple said alternation sets are associated uniquely with a single said indication.

8. A method of setting up a machine translation system comprising expanding in each language each of a set of word entries to each possible alternation thereof, using a consistent set of language-independent role data shared by words of each language with a common meaning.

9. A method of abstracting a language-dependent source language semantic structure to provide a language-independent semantic structure, using abstracting rules defining a set of alternations which words of the source language can take, a plurality of said words sharing each said set.

ST

ABSTRACT

A method of abstracting a language-dependent source language semantic structure to provide a language-independent semantic structure, using abstracting rules defining a set of alternations which words of the source language can take, a plurality of said words sharing each said set.

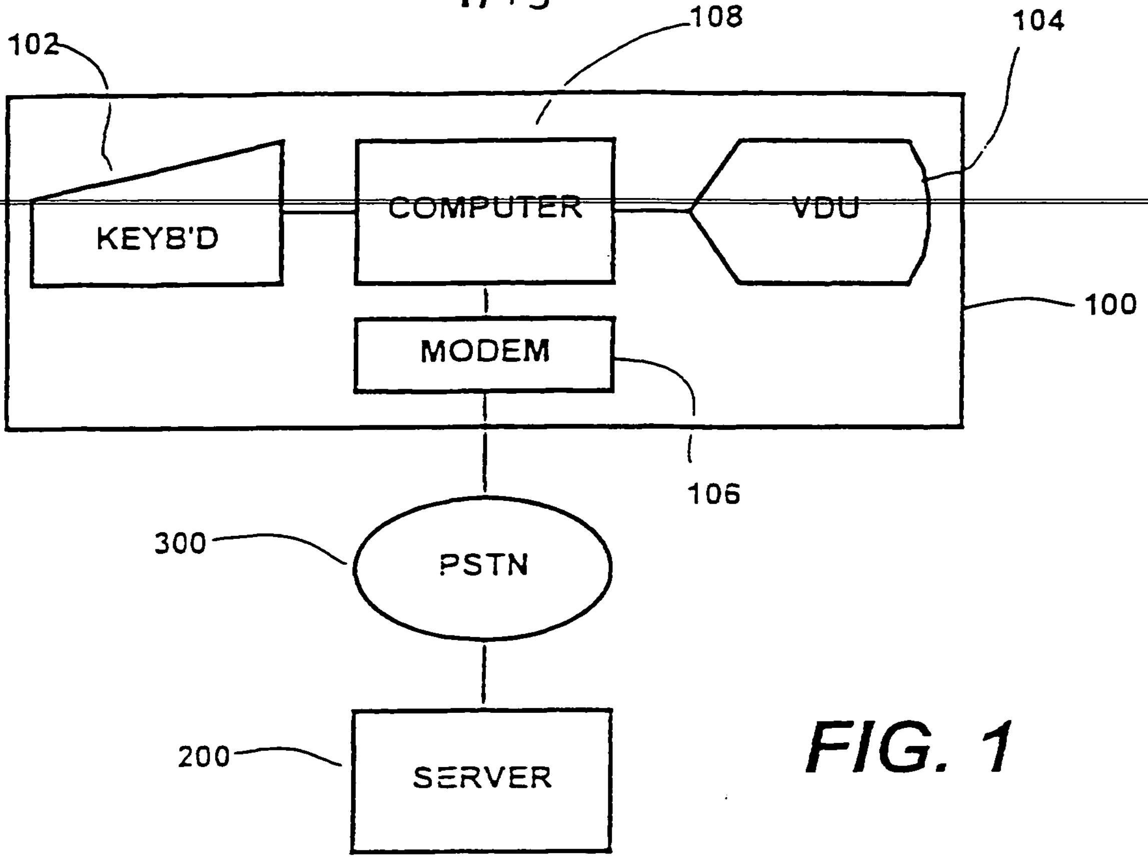

102
108
104
KEYB'D
COMPUTER
VDU
MODEM
100
106
300
PSTN
200
SERVER

FIG. 1

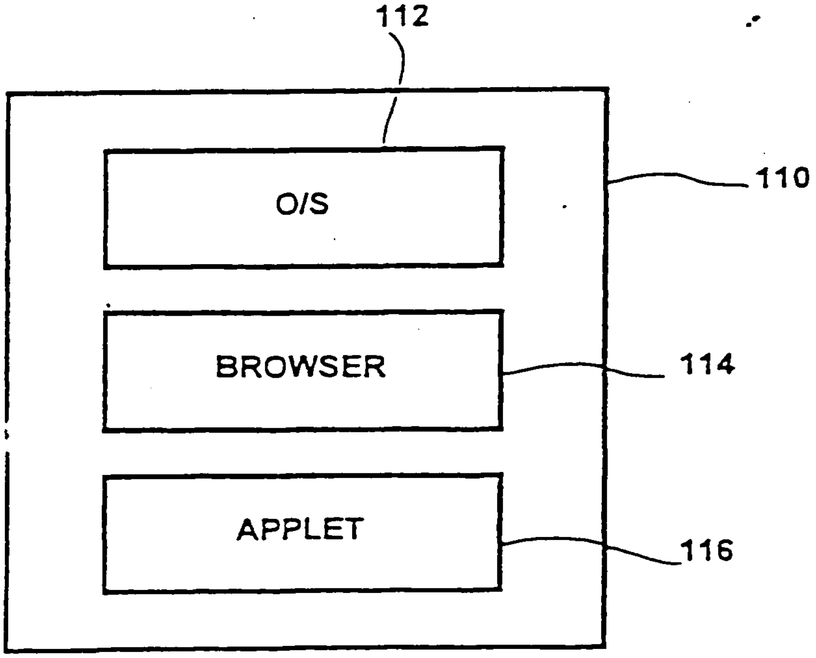
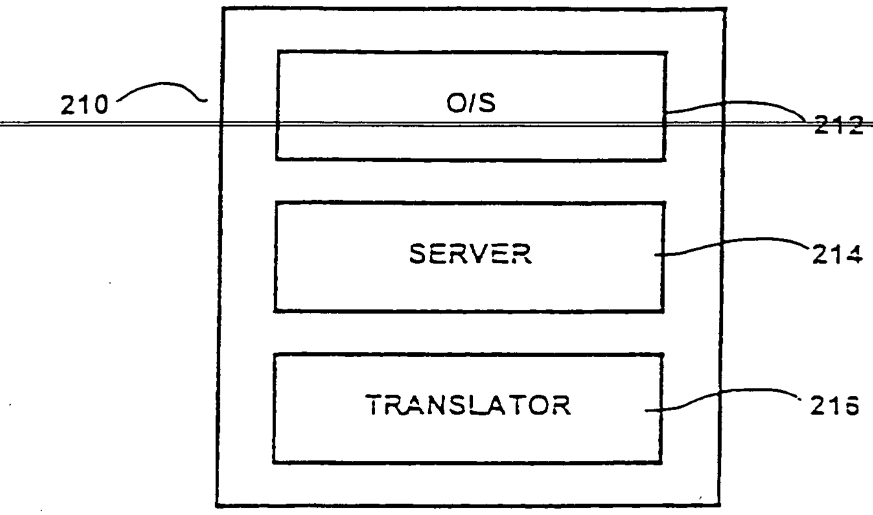
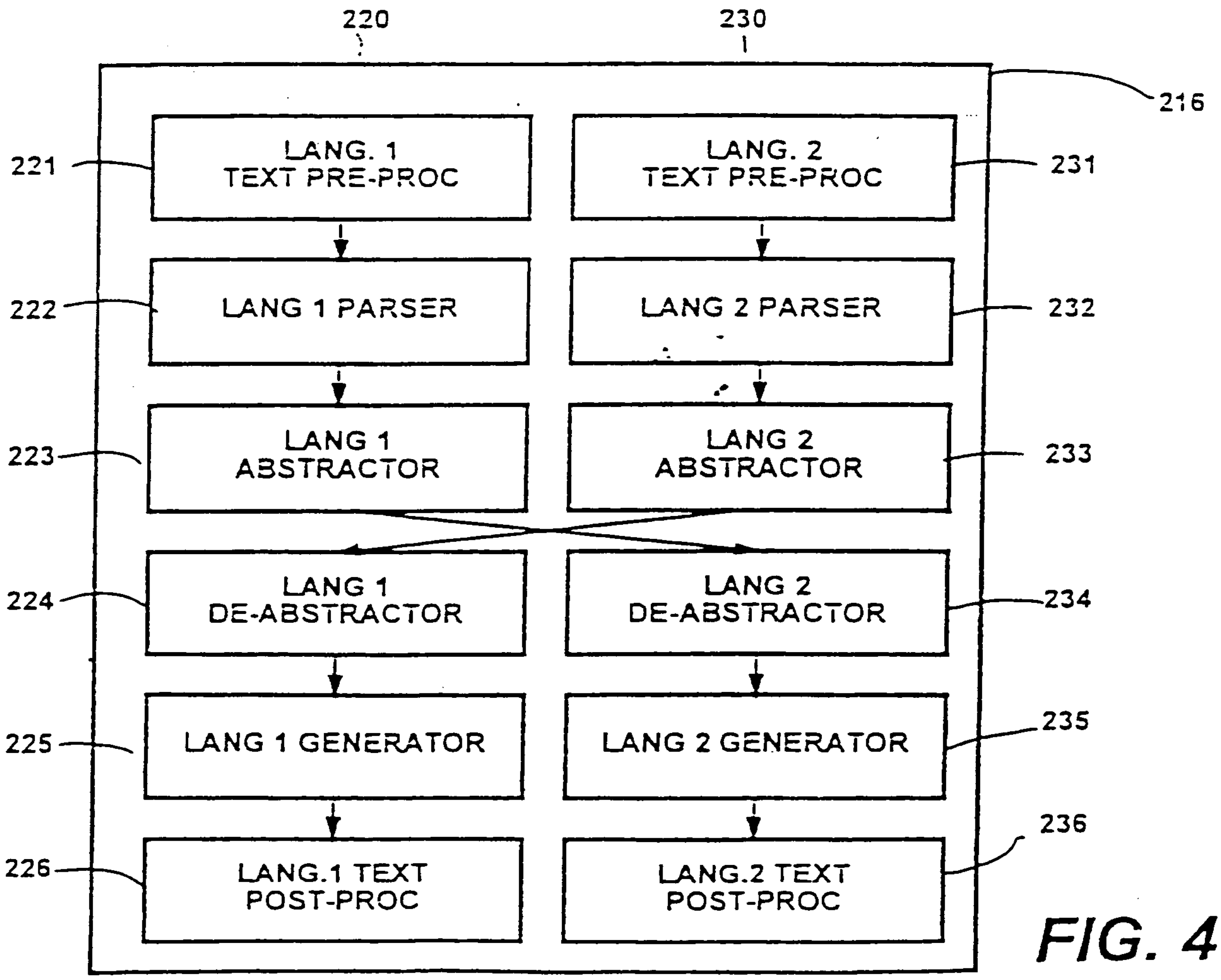

112
O/S
110
BROWSER
114
APPLET
116

FIG. 2



210
O/S
212
SERVER
214
TRANSLATOR
216

FIG. 3



220
230
216
221
LANG. 1 TEXT PRE-PROC
LANG. 2 TEXT PRE-PROC
231
222
LANG 1 PARSER
LANG 2 PARSER
232
223
LANG 1 ABSTRACTOR
LANG 2 ABSTRACTOR
233
224
LANG 1 DE-ABSTRACTOR
LANG 2 DE-ABSTRACTOR
234
225
LANG 1 GENERATOR
LANG 2 GENERATOR
235
226
LANG.1 TEXT POST-PROC
LANG.2 TEXT POST-PROC
236

FIG. 4

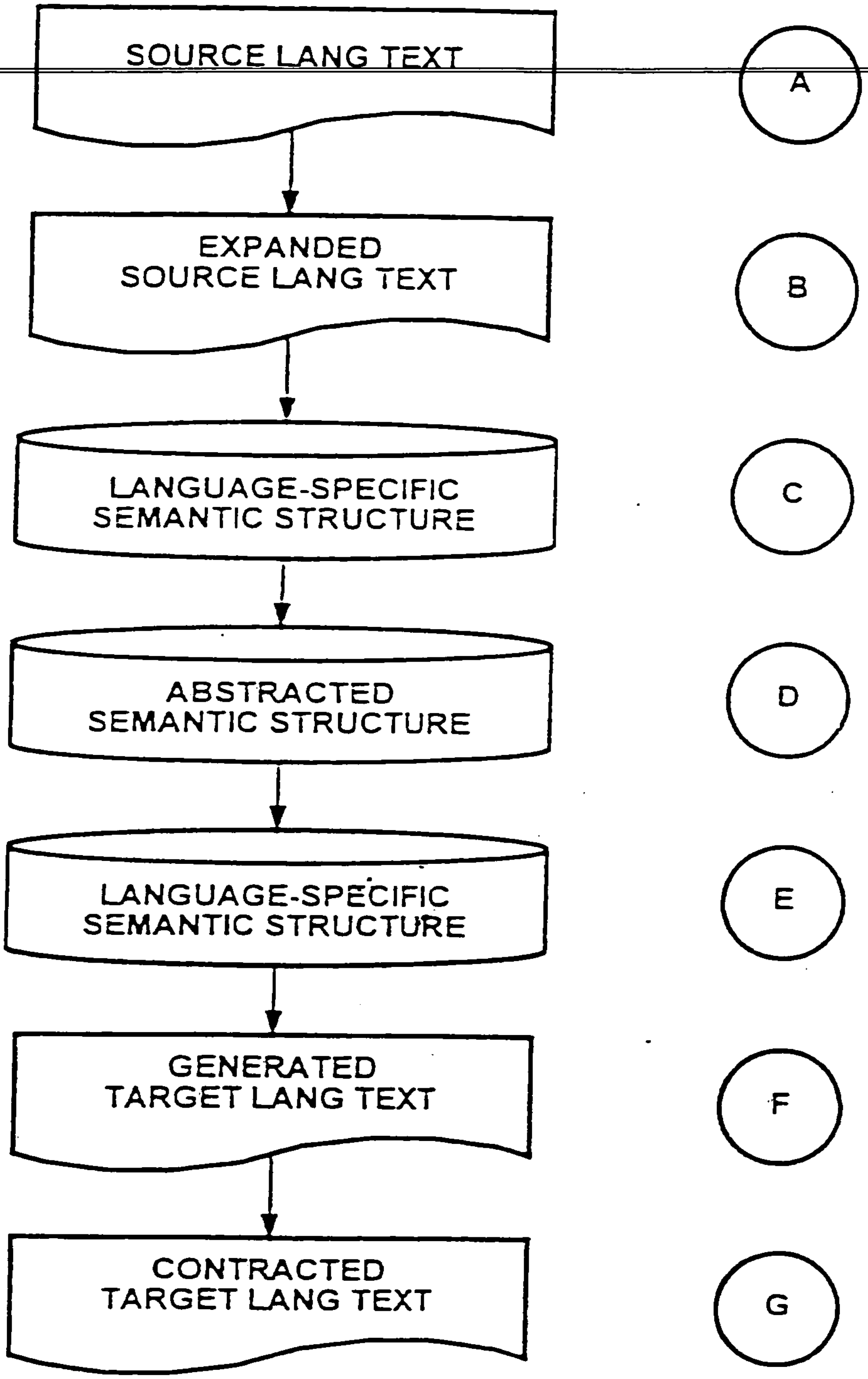

SOURCE LANG TEXT
A
EXPANDED SOURCE LANG TEXT
B
LANGUAGE-SPECIFIC SEMANTIC STRUCTURE
C
ABSTRACTED SEMANTIC STRUCTURE
D
LANGUAGE-SPECIFIC SEMANTIC STRUCTURE
E
GENERATED TARGET LANG TEXT
F
CONTRACTED TARGET LANG TEXT
G

FIG. 5

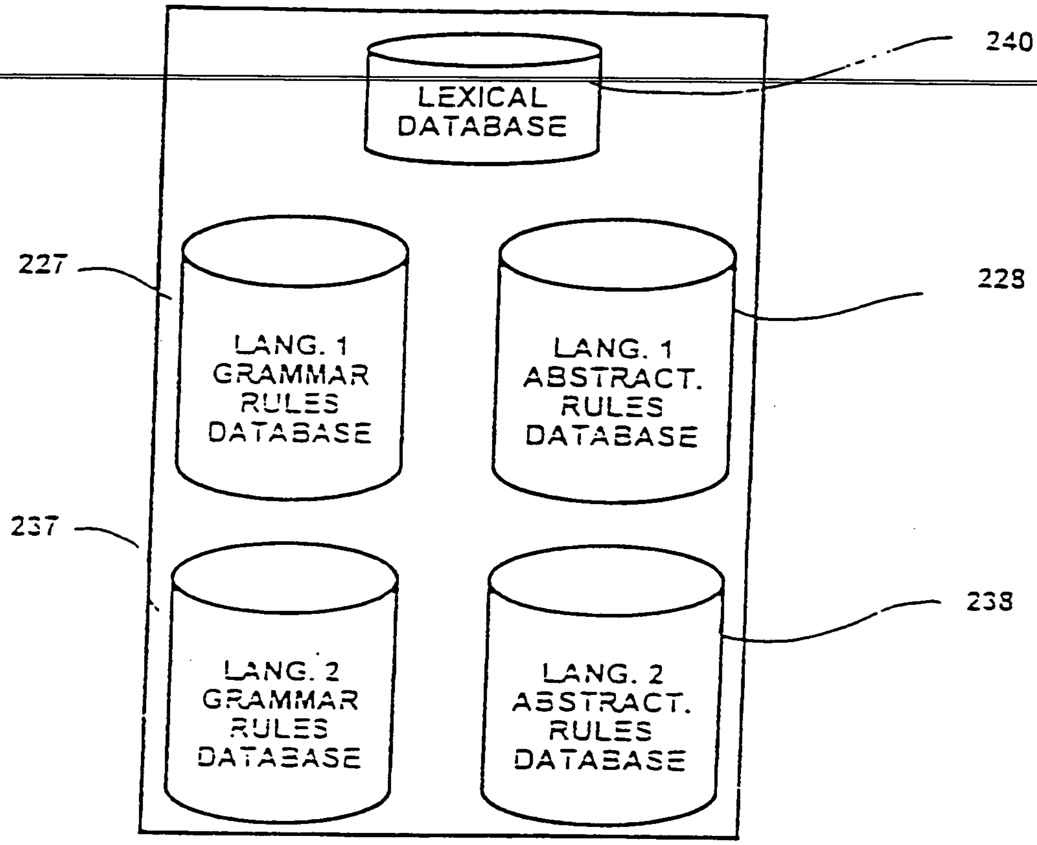

LEXICAL DATABASE
240
227
LANG. 1 GRAMMAR RULES DATABASE
LANG. 1 ABSTRACT. RULES DATABASE
228
237
LANG. 2 GRAMMAR RULES DATABASE
LANG. 2 ABSTRACT. RULES DATABASE
238

FIG. 6

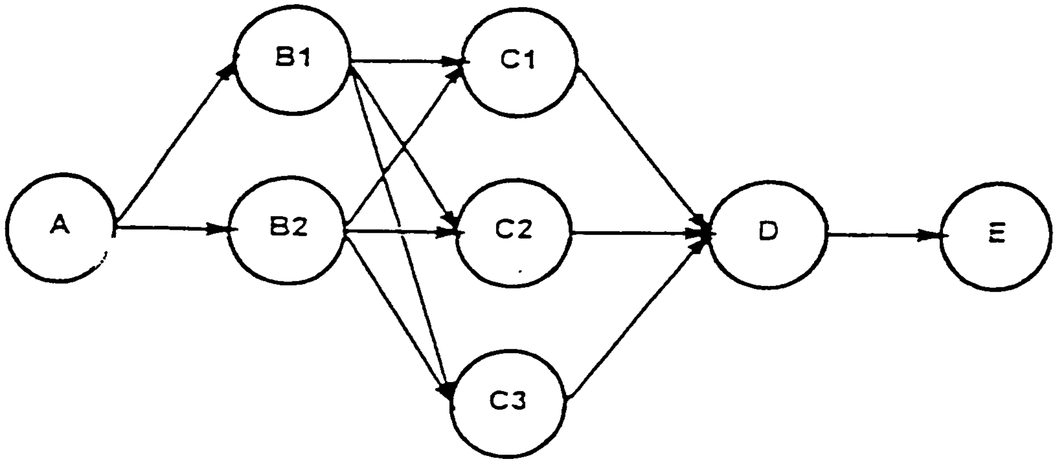

B1
C1
A
B2
C2
D
E
C3

FIG. 7

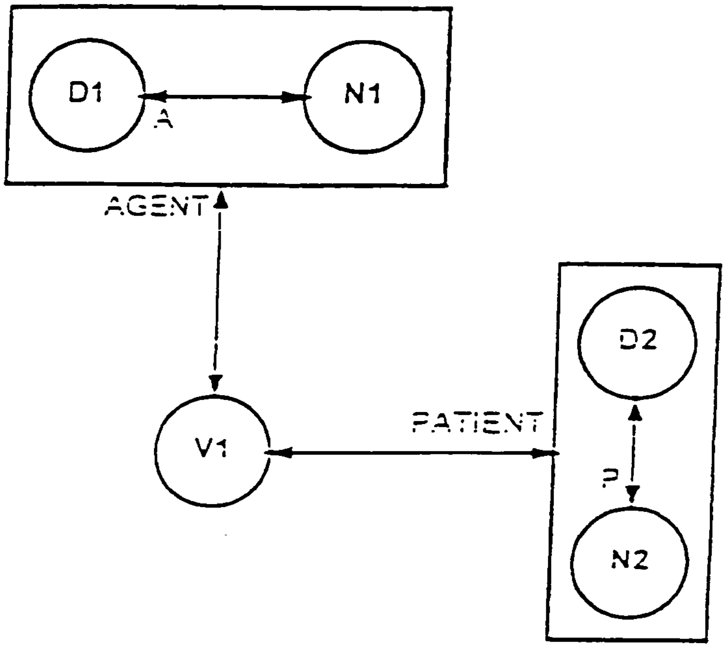


FIG. 8

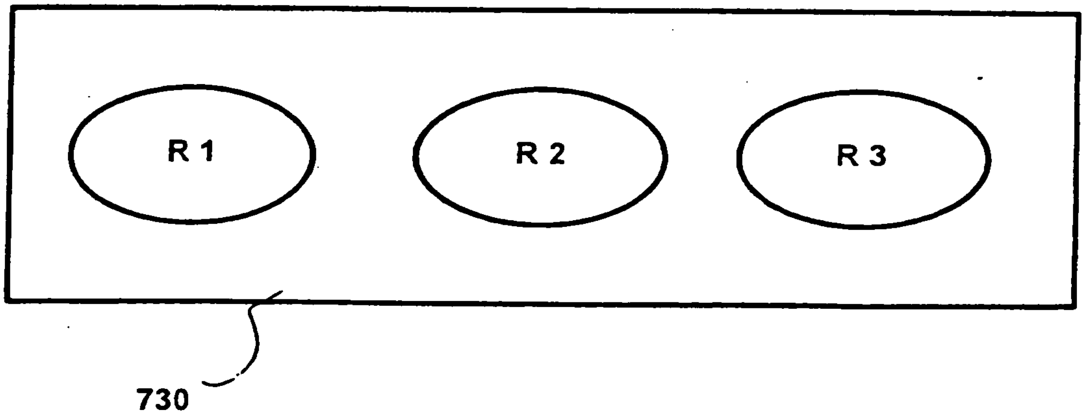


FIG. 17

6/13

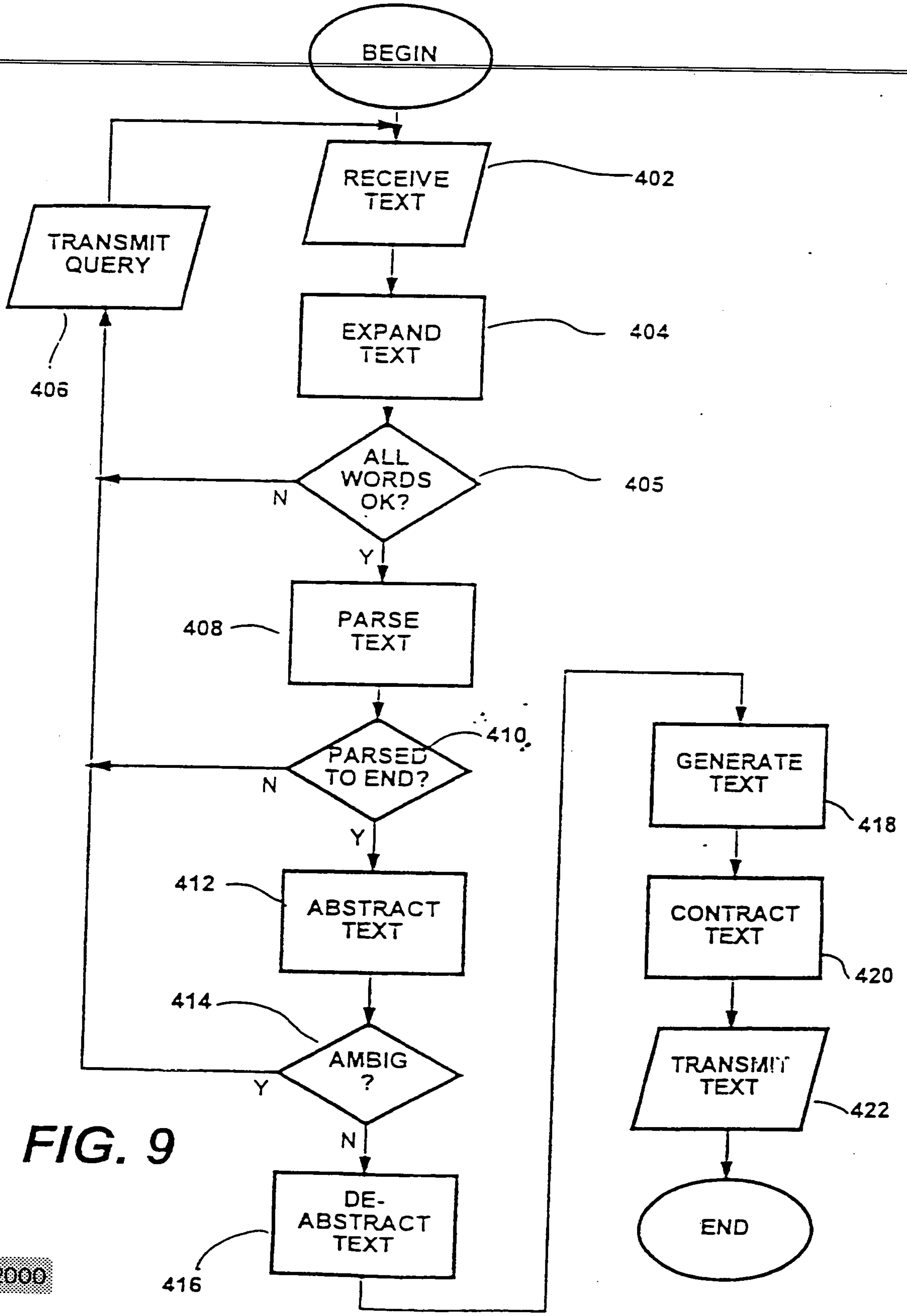

BEGIN
RECEIVE TEXT
402
TRANSMIT QUERY
406
EXPAND TEXT
404
ALL WORDS OK?
405
N
Y
PARSE TEXT
408
PARSED TO END?
410
N
Y
ABSTRACT TEXT
412
414
AMBIG ?
Y
N
DE-ABSTRACT TEXT
416
GENERATE TEXT
418
CONTRACT TEXT
420
TRANSMIT TEXT
422
END

FIG. 9

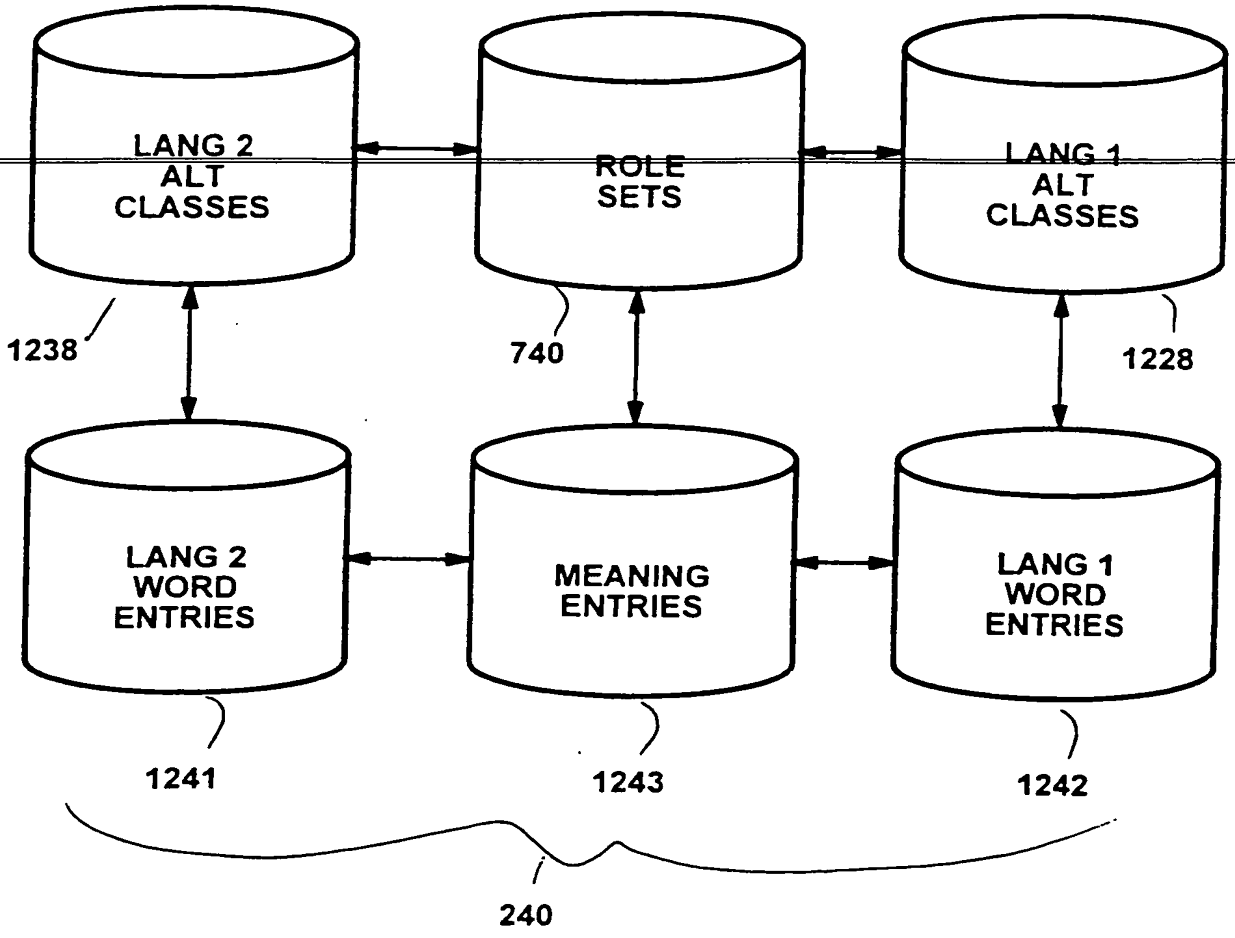

LANG 2 ALT CLASSES
ROLE SETS
LANG 1 ALT CLASSES
1238
740
1228
LANG 2 WORD ENTRIES
MEANING ENTRIES
LANG 1 WORD ENTRIES
1241
1243
1242
240

FIG. 10

8/13

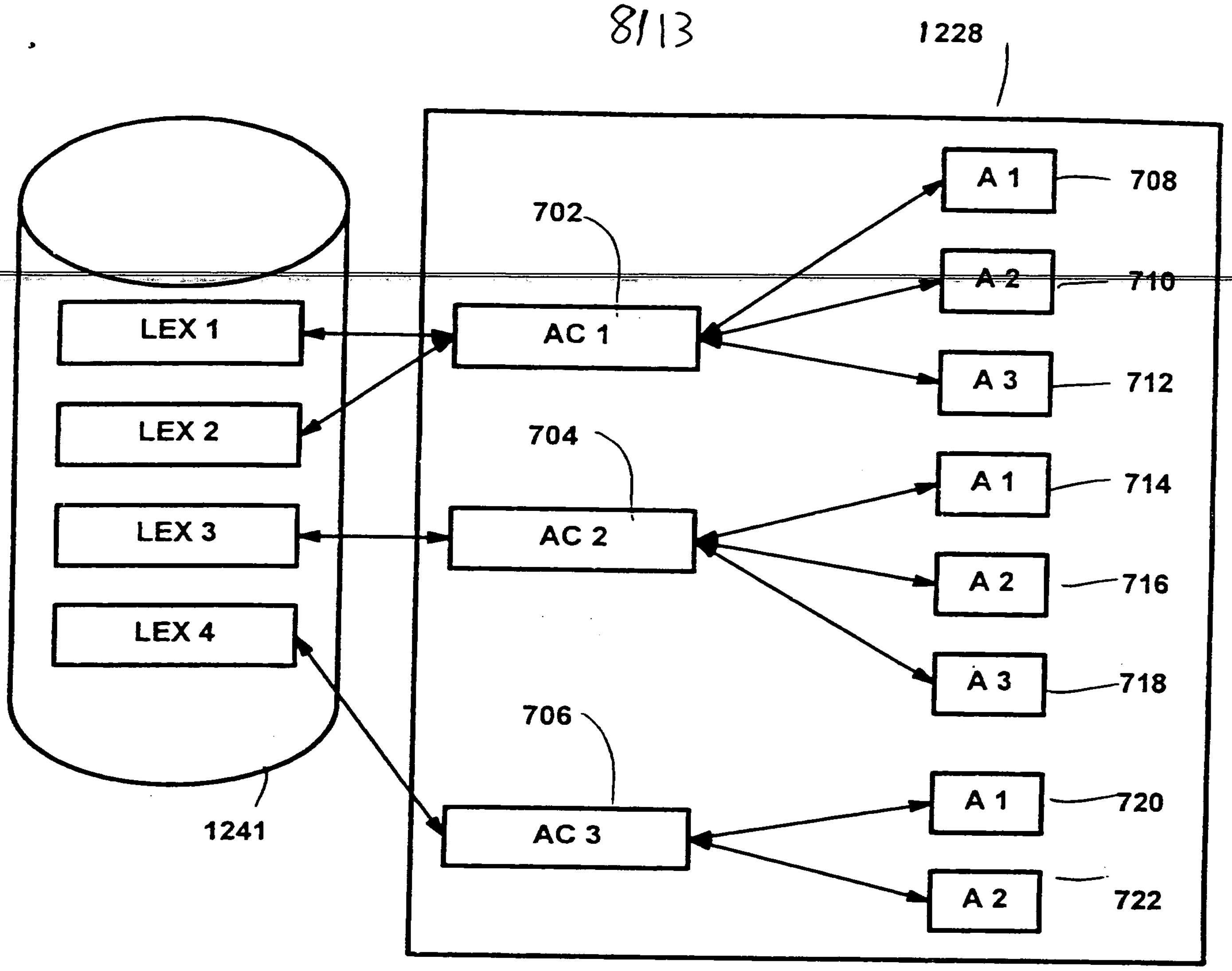

1228
LEX 1
LEX 2
LEX 3
LEX 4
1241
702
AC 1
704
AC 2
706
AC 3
A 1
708
A 2
710
A 3
712
A 1
714
A 2
716
A 3
718
A 1
720
A 2
722

FIG. 11

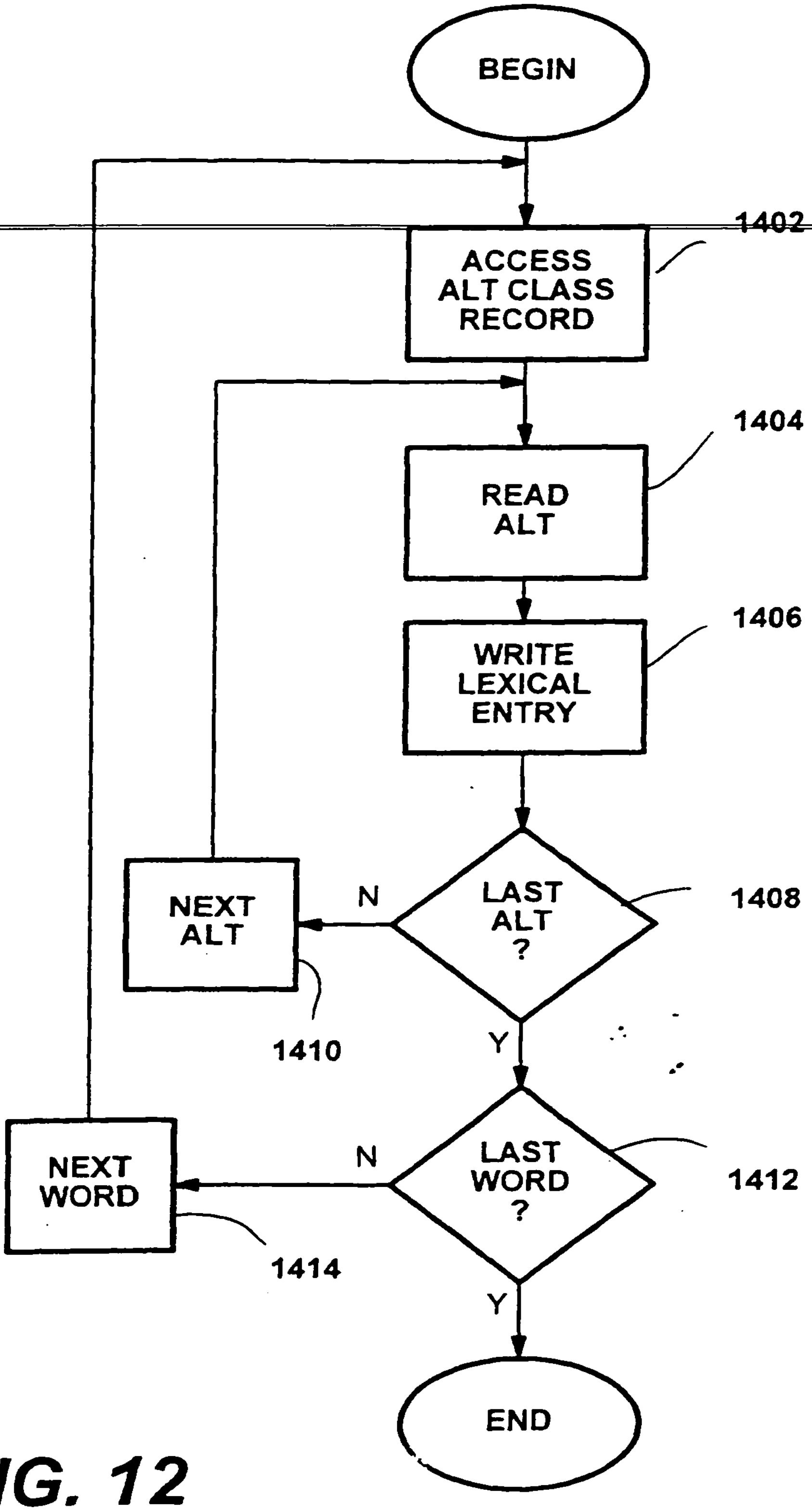

BEGIN
1402
ACCESS ALT CLASS RECORD
1404
READ ALT
1406
WRITE LEXICAL ENTRY
LAST ALT ?
1408
N
NEXT ALT
1410
Y
LAST WORD ?
1412
N
NEXT WORD
1414
Y
END

FIG. 12

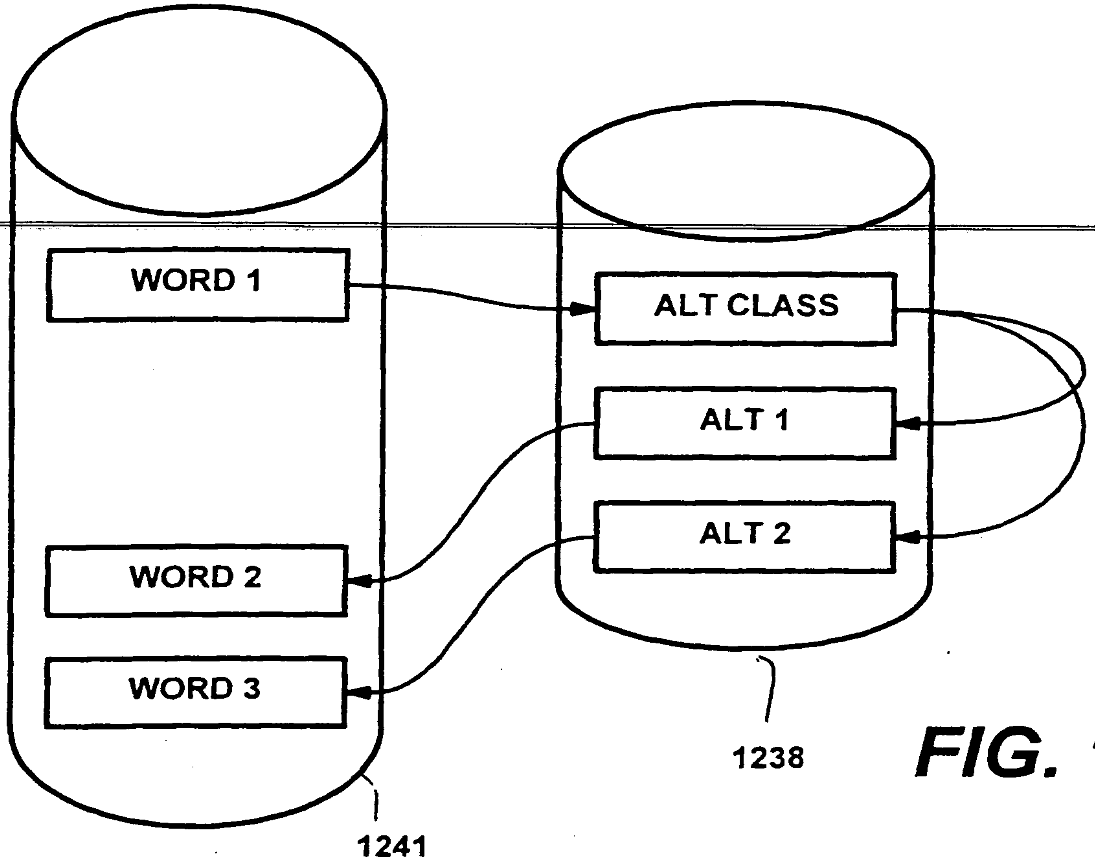


FIG. 13

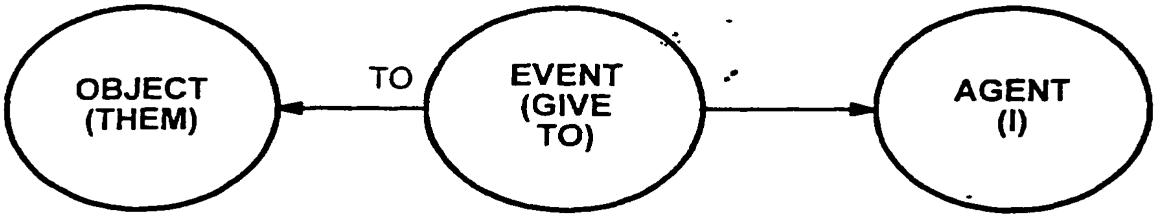


FIG. 14a

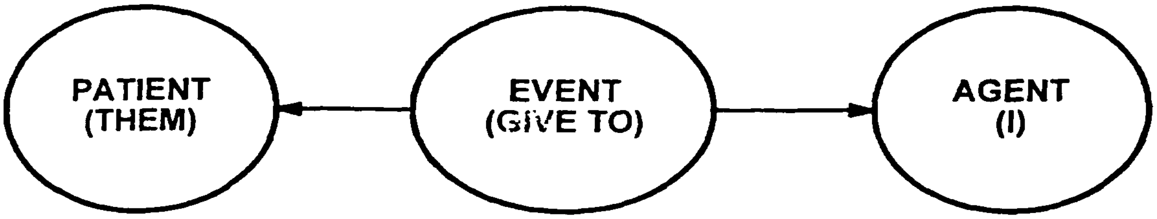
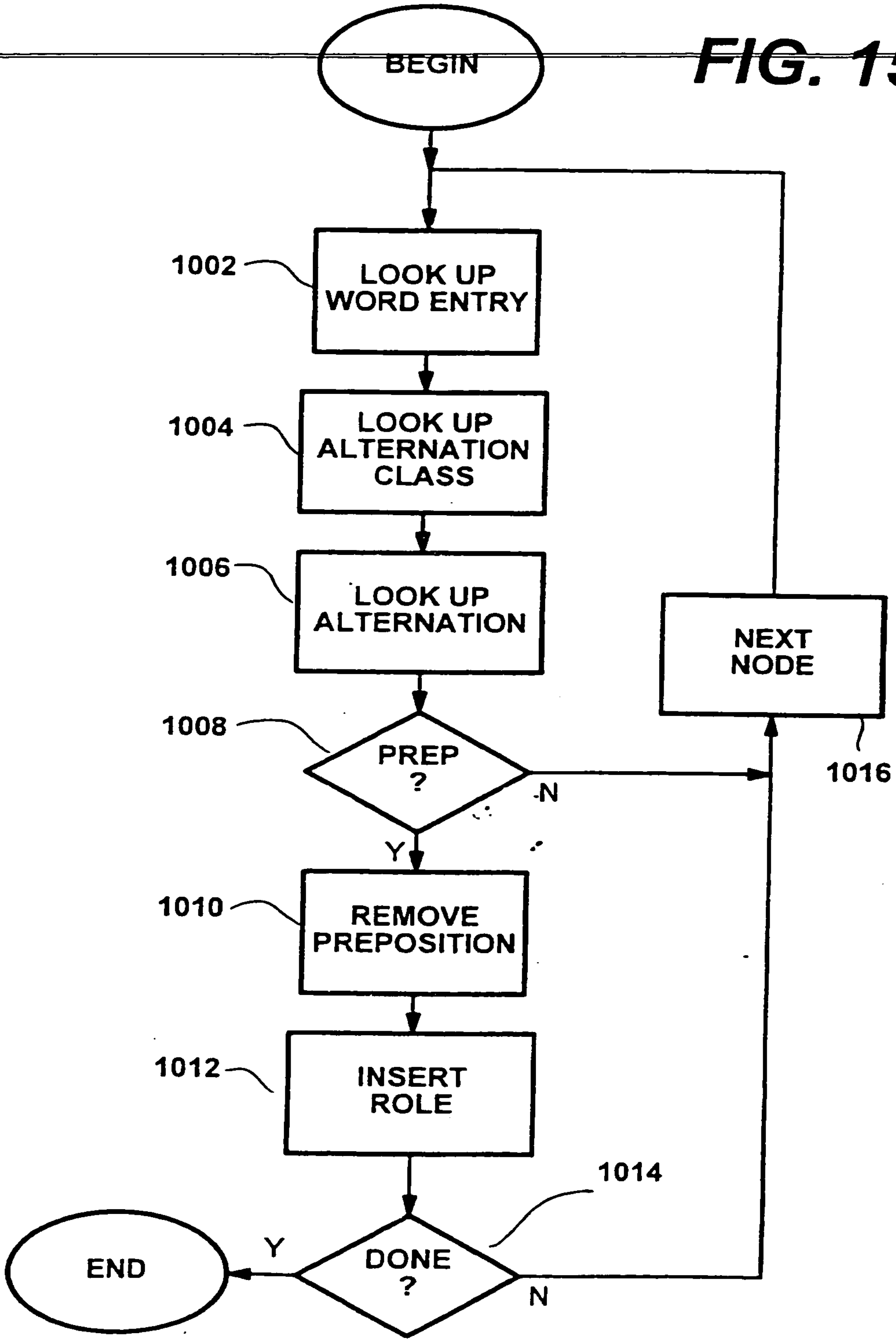


FIG. 14b

FIG. 15


BEGIN
1002
LOOK UP WORD ENTRY
1004
LOOK UP ALTERNATION CLASS
1006
LOOK UP ALTERNATION
1008
PREP ?
N
Y
NEXT NODE
1016
1010
REMOVE PREPOSITION
1012
INSERT ROLE
1014
DONE ?
Y
N
END

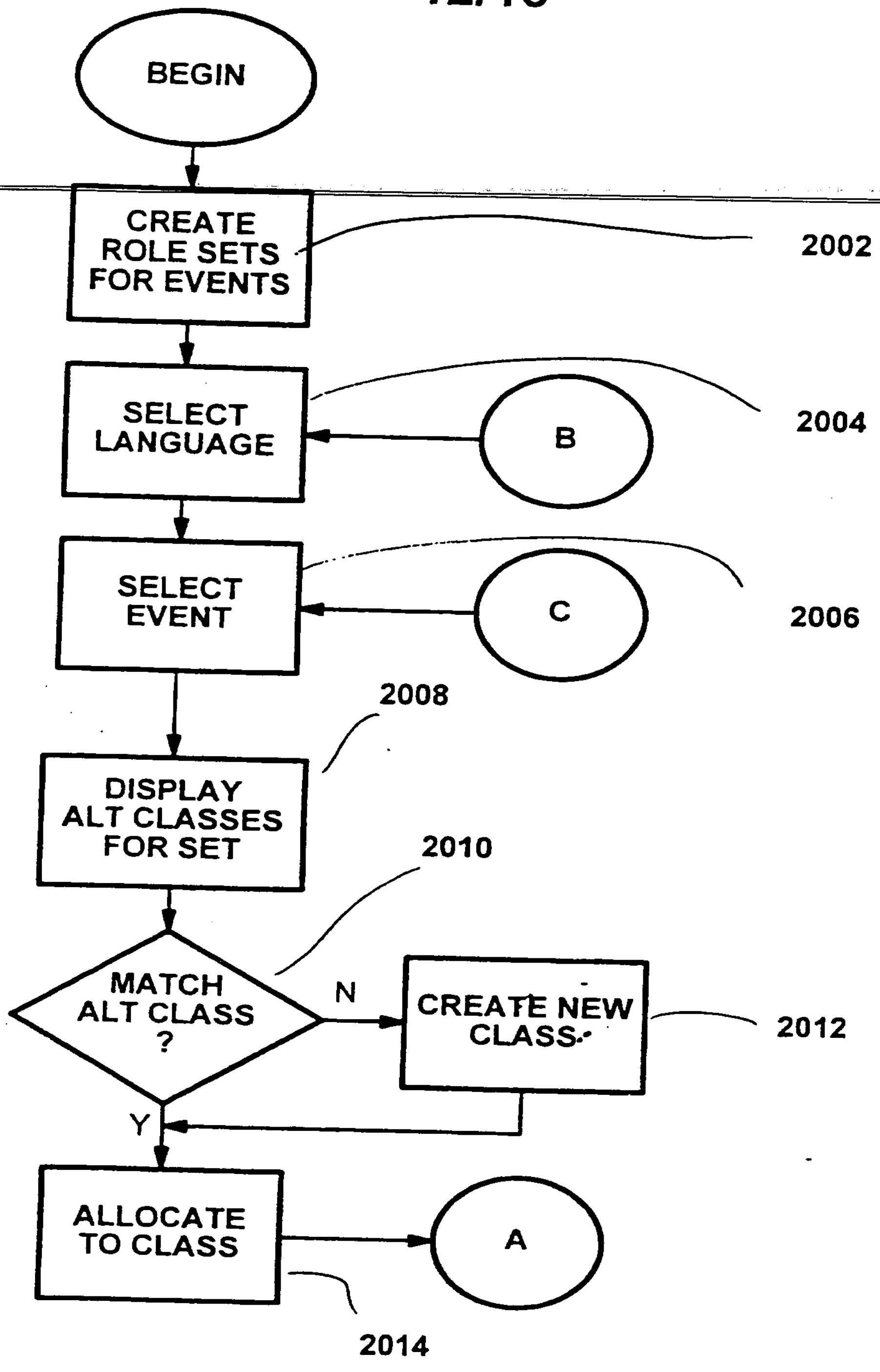

BEGIN
CREATE ROLE SETS FOR EVENTS
2002
SELECT LANGUAGE
B
2004
SELECT EVENT
C
2006
2008
DISPLAY ALT CLASSES FOR SET
2010
MATCH ALT CLASS ?
N
CREATE NEW CLASS
2012
Y
ALLOCATE TO CLASS
A
2014

FIG. 16a

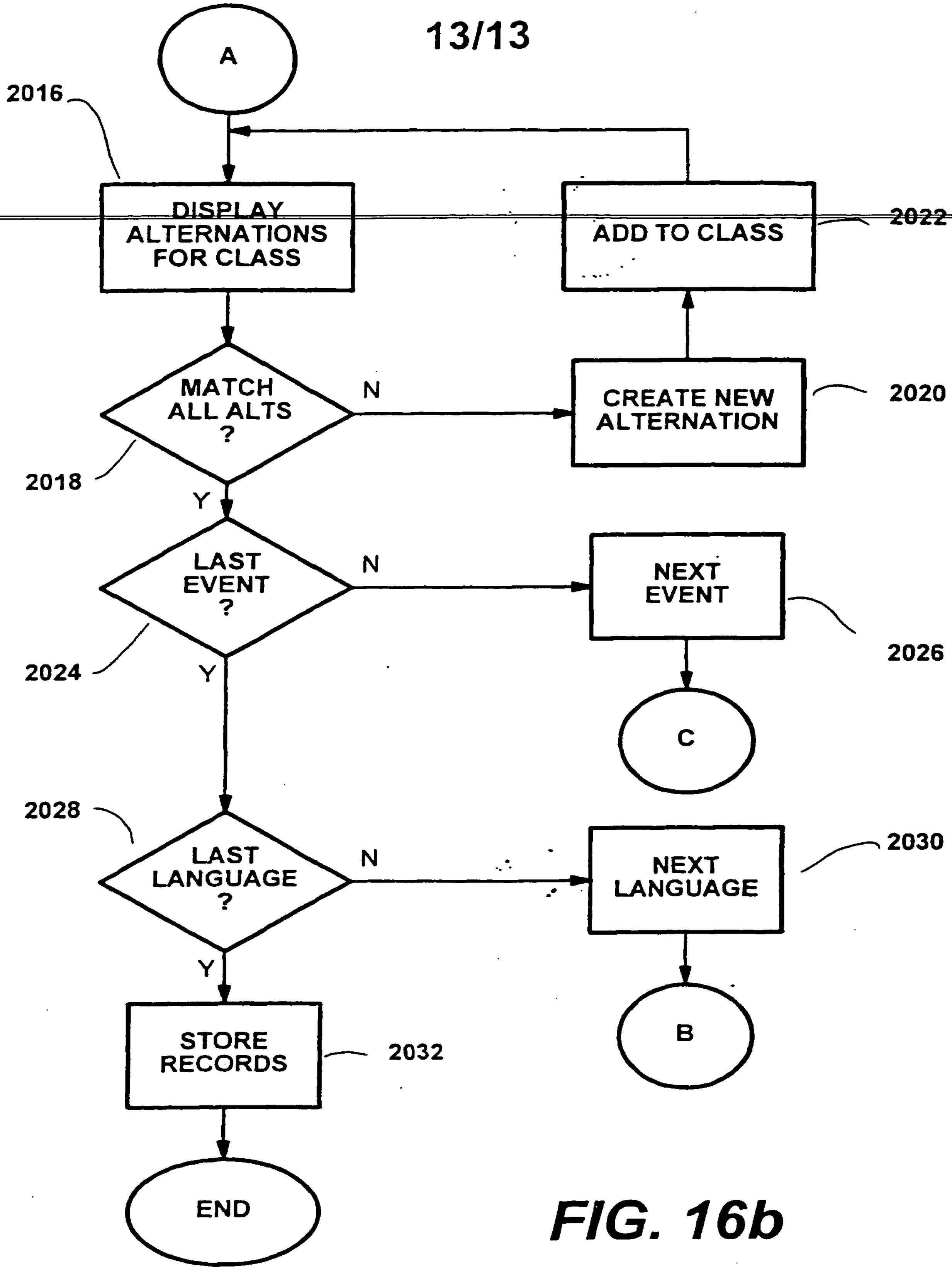

A
2016
DISPLAY ALTERNATIONS FOR CLASS
ADD TO CLASS
2022
MATCH ALL ALTS ?
N
CREATE NEW ALTERNATION
2020
2018
Y
LAST EVENT ?
N
NEXT EVENT
2026
2024
Y
C
2028
LAST LANGUAGE ?
N
NEXT LANGUAGE
2030
Y
STORE RECORDS
2032
B
END

FIG. 16b